2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Inverse Dynamics Control with Floating Base and Constraints

Jun Nakanishi[*†], Michael Mistry[‡], and Stefan Schaal[†‡]
[*] *ICORP, Japan Science and Technology Agency, Kawaguchi, Saitama, Japan*
[†] *ATR Computational Neuroscience Laboratories, Kyoto, Japan*
[‡] *Computer Science and Neuroscience, University of Southern California, Los Angeles, USA*
jun@atr.jp, {mmistry, sschaal}@usc.edu

Abstract— **In this paper, we address the issues of compliant control of a robot under contact constraints with a goal of using joint space based pattern generators as movement primitives, as often considered in the studies of legged locomotion and biological motor control. For this purpose, we explore inverse dynamics control of constrained dynamical systems. When the system is overconstrained, it is not straightforward to formulate an inverse dynamics control law since the problem becomes an ill-posed one, where infinitely many combinations of joint torques are possible to achieve the desired joint accelerations.**

The goal of this paper is to develop a general and computationally efficient inverse dynamics algorithm for a robot with a free floating base and constraints. We suggest an approximate way of computing inverse dynamics algorithm by treating constraint forces computed with a Lagrange multiplier method as simply external forces based on Featherstone's floating base formulation of inverse dynamics. We present how all the necessary quantities to compute our controller can be efficiently extracted from Featherstone's spatial notation of robot dynamics. We evaluate the effectiveness of the suggested approach on a simulated biped robot model.

I. Introduction

Movement tasks are defined in either task space or configuration space. For instance, in reaching for an object, task coordinates (e.g., Cartesian coordinates) usually play the dominant role for successful control, while in legged locomotion, pattern generation in joint space seems to be of high importance. For systems with redundant degrees-of-freedom, even task space control has usually an additional configuration space component due to some form of null space control [1]. Thus, besides accurate task space control, most commonly addressed in the framework of operational space control [2], [3], there is also the need for accurate configuration space control.

Accurate control can be accomplished with many different control schemes. But if we wish to achieve a high level of compliance, e.g., as needed for the new wave of humanoid, entertainment, and assistive robots for safe operation in human environments [4], model-based control seems to be the only viable approach to keep feedback gains low without compromising tracking accuracy. Model-based control in task space and in joint space is well understood as long as the robot is a fixed base system with no additional constraints. However, if constraints exist, as in walking robots or robots interacting with rigid objects in the environment (e.g., when tracing a rigid surface or writing on a board), model-based control becomes more complex. The most common approach for model-based control with constraints is to derive the constrained dynamics, and subsequently, suitable control laws. For instance, in [5], [6], some general principles are given how to approach such constrained dynamics problems. As long as the number of task dimensions is smaller than the dimensionality of the constrained dynamical system, the control system is usually feedback linearizable and thus suitable for model-based control.

However, as mentioned above, in many robot tasks, configuration space control is an important part of the control objective. Biped locomotion with joint-space pattern generators [7] is a typical example. For instance, consider a biped robot with 7 degrees-of-freedom (DOFs) per leg. This system has a total of 20 DOFs including the base 6 DOFs. In the double stance phase, each foot creates a six-dimensional constraint, such that the constrained dynamics has effectively 8 DOFs. Thus, standard feedback linearization for model-based control is not possible for this system as it is overconstraint, such that it is unclear how ideas of joint-space pattern generators for locomotion can be combined with compliant model-based control.

In this paper, we will address exactly this problem, i.e., how to realize compliant control of a robot under contact constraints when our goal is to use joint space based pattern generators, e.g., as in the movement primitive approach used in [7]. For this purpose, we will explore inverse dynamics control of constrained dynamical systems. Robot control with kinematic constraints has been extensively studied in object manipulation with a robotic hand or multiple robot arms [8], [9], position and force control of robot manipulators with constraints [9]–[13], modelling and control of a planar legged robot under friction constraints [14], and humanoid robot control [15], [16]. Inverse dynamics control computes the motor commands (joint torques) for the given desired joint accelerations. If the control system is overconstraint, infinitely many combinations of joint torques are possible to achieve the desired joint accelerations, where each particular realization creates a different set of constraint/contract forces. Furthermore, since the constraint forces depend on the current joint torques, it is not straightforward to determine the joint torques which achieve the desired joint accelerations by simply compensating for the nonlinear robot dynamics and the necessary constraint forces.

The goal of this paper is to develop a general and computationally efficient inverse dynamics algorithm for constrained

1-4244-0602-1/07/$20.00 ©2007 IEEE.

dynamics starting from the formulation of the robot as a free floating base system. First, given the floating base rigid body dynamics, it is possible to extract the constrained joint space dynamics for arbitrary constraints from a Lagrange multiplier approach and Khatib's framework of operational space control [10]. In general, the constrained joint space dynamics cannot be solved easily for the appropriate torques given a (constraint compatible) desired trajectory. Second, we consider an approximate way of computing the inverse dynamics. Using Featherstone's floating base formulation of inverse dynamics [17] with external forces, we treat the Lagrange multipliers simply as such external forces, which can be justified as an approximation to the true contact forces at high sampling rates and smooth motor command generation. Third, we consider how all needed quantities for our control approach can be computed efficiently from Featherstone's spatial notation of robot dynamics [17]. At last, we evaluate the suggested approach on a simulated biped robot model with several benchmark movements.

II. Problem Setup

Consider the floating base rigid body dynamics, i.e., the combined dynamics of generalized coordinates of the robot and the six DOFs added by the floating base:

$$\tilde{\mathbf{M}}(\tilde{\mathbf{q}})\ddot{\tilde{\mathbf{q}}} + \tilde{\mathbf{C}}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) + \tilde{\mathbf{g}}(\mathbf{q}) = \mathbf{S}^T\boldsymbol{\tau} \tag{1}$$

$\tilde{\mathbf{q}} = [\mathbf{q}^T, \mathbf{x}_B^T]^T$ denotes the entire robot position vector composed of the joint angle vector $\mathbf{q}$ and the six base DOFs $\mathbf{x}_B$, and, in general, all quantities referring to the full floating base dynamics are characterized by the "˜" above variables. Thus, in (1), $\tilde{\mathbf{M}}(\tilde{\mathbf{q}})$ is the floating base inertia matrix, $\tilde{\mathbf{C}}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})$ is the floating base Coriolis/centripetal vector, $\tilde{\mathbf{g}}(\tilde{\mathbf{q}})$ is the floating base gravity vector, and $\boldsymbol{\tau}$ is the joint torque vector. We use the selection matrix $\mathbf{S} = [\mathbf{I}_{n\times n}, \mathbf{0}_{6\times 6}]$ to extract joint space variables from the full floating base variables.

With kinematic constraints of the form $\tilde{\mathbf{h}}_c(\tilde{\mathbf{q}}) = 0$, the constrained dynamics become

$$\tilde{\mathbf{M}}\ddot{\tilde{\mathbf{q}}} + \tilde{\mathbf{C}} + \tilde{\mathbf{g}} = \mathbf{S}^T\boldsymbol{\tau} + \tilde{\mathbf{J}}_c^T\boldsymbol{\lambda} \tag{2}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multiplier describing the constraint forces, and $\tilde{\mathbf{J}}_c$ is the Jacobian of $\tilde{\mathbf{h}}_c$. Figure 1 depicts an example of a robot in contact with the environment. From a Lagrange multiplier approach, the constraint forces $\boldsymbol{\lambda}$ can be computed, e.g., as [8], [18]

$$\boldsymbol{\lambda} = \left(\tilde{\mathbf{J}}_c\tilde{\mathbf{M}}^{-1}\tilde{\mathbf{J}}_c^T\right)^{-1}\left[\tilde{\mathbf{J}}_c\tilde{\mathbf{M}}^{-1}\left(\tilde{\mathbf{C}} + \tilde{\mathbf{g}} - \mathbf{S}^T\boldsymbol{\tau}\right) - \dot{\tilde{\mathbf{J}}}_c\dot{\tilde{\mathbf{q}}}\right] \tag{3}$$

Note that the constraint forces depend on the current motor command $\boldsymbol{\tau}$. By substituting (3) into (2) and projecting the entire dynamics into the joint space, i.e., by eliminating the uncontrolled base DOFs, the constrained joint space dynamics can be extracted as [19]

$$\begin{aligned}\left(\mathbf{S}\tilde{\mathbf{M}}^{-1}\mathbf{S}^T\right)^{-1}\ddot{\mathbf{q}} + \bar{\mathbf{S}}^T\left(\mathbf{I} - \tilde{\mathbf{J}}_c^T\bar{\tilde{\mathbf{J}}}_c^T\right)\left(\tilde{\mathbf{C}} + \tilde{\mathbf{g}}\right)&\\ = \bar{\mathbf{S}}^T\left(\mathbf{I} - \tilde{\mathbf{J}}_c^T\bar{\tilde{\mathbf{J}}}_c^T\right)\mathbf{S}^T\boldsymbol{\tau}&\end{aligned} \tag{4}$$

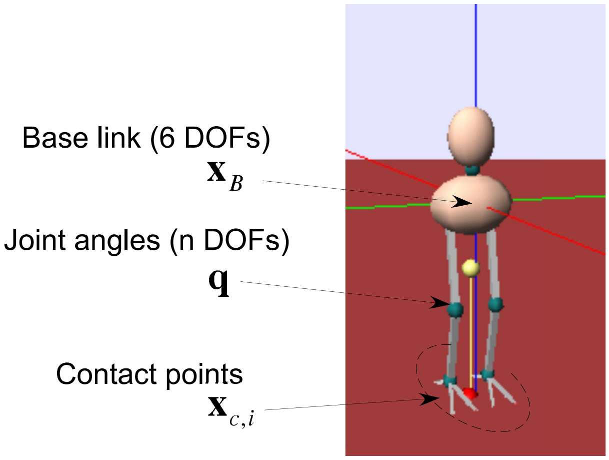


Fig. 1. Example of a robot in contact with an environment.

where $\bar{\mathbf{S}}$ and $\bar{\tilde{\mathbf{J}}}_c$ are the inertia-weighted pseudoinverses of $\mathbf{S}$ and $\tilde{\mathbf{J}}_c$, defined as $\bar{\mathbf{S}} = \tilde{\mathbf{M}}^{-1}\mathbf{S}^T\left(\mathbf{S}\tilde{\mathbf{M}}^{-1}\mathbf{S}^T\right)^{-1}$ and $\bar{\tilde{\mathbf{J}}}_c = \tilde{\mathbf{M}}^{-1}\tilde{\mathbf{J}}_c^T\left(\tilde{\mathbf{J}}_c\tilde{\mathbf{M}}^{-1}\tilde{\mathbf{J}}_c^T\right)^{-1}$, respectively.

The problem with the inverse dynamics of the constrained joint space dynamics (4) is that, depending on the form and number of the constraints, the matrix $\bar{\mathbf{S}}^T(\mathbf{I} - \tilde{\mathbf{J}}_c^T\bar{\tilde{\mathbf{J}}}_c^T)\mathbf{S}^T$ is not always invertible. Several ideas exist to deal with this issue. First, additional optimization criteria could be introduced to obtain a proper solution to the inversion of (4). This approach, however, creates very task and robot specific solutions, and does not easily deal with cases where constraints may dynamically appear as the robot touches the world with its feet, arms, or other body parts. Second, as mentioned in [18], if we could project the floating base dynamics onto the correct reduced rank space of the constraint dynamics, the inverse dynamics problem could be solved by a simple pseudo-inverse method. However, this projection is generally not known and might only be obtained by means of numerical methods—from our experience, this approach was numerically not very robust. Thus, in the next section, we consider an alternative approach to computing inverse dynamics of the constrained dynamical system by treating $\boldsymbol{\lambda}$ simply as external forces.

III. Controller Formulations

The difficulty of computing inverse dynamics with constraints is that the constraint forces depend on the current motor command for which we would like to solve the equations. Suppose we treat $\boldsymbol{\lambda}$ as given external forces $\mathbf{F}_{ext}$. Then the joint space dynamics becomes:

$$\left(\mathbf{S}\tilde{\mathbf{M}}^{-1}\mathbf{S}^T\right)^{-1}\ddot{\mathbf{q}} + \bar{\mathbf{S}}^T\left(\tilde{\mathbf{C}} + \tilde{\mathbf{g}} - \tilde{\mathbf{J}}_c^T\mathbf{F}_{ext}\right) = \boldsymbol{\tau}. \tag{5}$$

In this case, the inverse dynamics problem is now well-posed such that we can determine the joint torques for the given joint accelerations. Thus, instead of solving (4) for $\boldsymbol{\tau}$, we consider an approximate way of computing inverse dynamics. The basic idea revolves around the assumption that the motor commands $\boldsymbol{\tau}$ from the previous time step are very close to the motor commands computed at the current time step. Under this assumption, we compute the constraint forces using $\boldsymbol{\tau}_{t-1}$

$$\begin{aligned}&\boldsymbol{\lambda}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}, \boldsymbol{\tau}^{t-1})\\&= \left(\tilde{\mathbf{J}}_c\tilde{\mathbf{M}}^{-1}\tilde{\mathbf{J}}_c^T\right)^{-1}\left[\tilde{\mathbf{J}}_c\tilde{\mathbf{M}}^{-1}\left(\tilde{\mathbf{C}} + \tilde{\mathbf{g}} - \mathbf{S}^T\boldsymbol{\tau}_{t-1}\right) - \dot{\tilde{\mathbf{J}}}_c\dot{\tilde{\mathbf{q}}}\right] \quad (6)\end{aligned}$$

The complete control law can be formulated as

$$\boldsymbol{\tau} = \left(\mathbf{S}\tilde{\mathbf{M}}^{-1}\mathbf{S}^T\right)^{-1}\ddot{\mathbf{q}} + \bar{\mathbf{S}}^T\left(\tilde{\mathbf{C}} + \tilde{\mathbf{g}} - \tilde{\mathbf{J}}_c^T\boldsymbol{\lambda}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}, \boldsymbol{\tau}_{t-1})\right). \quad (7)$$

The same control law can be derived in a different way, from the idea of hybrid dynamics algorithm in [17], in which, for the given desired joint accelerations, the base accelerations and joint torques are computed. First, the robot dynamics equation

$$\begin{bmatrix} \tilde{\mathbf{M}}_{11} & \tilde{\mathbf{M}}_{12} \\ \tilde{\mathbf{M}}_{21} & \tilde{\mathbf{M}}_{22} \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{q}} \\ \ddot{\mathbf{x}}_B \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{C}}_1 + \tilde{\mathbf{g}_1} \\ \tilde{\mathbf{C}}_2 + \tilde{\mathbf{g}_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} \\ 0 \end{bmatrix} + \tilde{\mathbf{J}}_c^T\mathbf{F}_{ext} \quad (8)$$

is rearranged as

$$\begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{M}}_{12} \\ \mathbf{0} & -\tilde{\mathbf{M}}_{22} \end{bmatrix}\begin{bmatrix} \boldsymbol{\tau} \\ \ddot{\mathbf{x}}_B \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{M}}_{11}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}_1 + \tilde{\mathbf{g}_1} - \tilde{\mathbf{J}}_{c1}^T\mathbf{F}_{ext} \\ \tilde{\mathbf{M}}_{21}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}_2 + \tilde{\mathbf{g}_2} - \tilde{\mathbf{J}}_{c2}^T\mathbf{F}_{ext} \end{bmatrix}. \quad (9)$$

Subsequently, we can solve for the joint torques $\boldsymbol{\tau}$ and the base accelerations $\ddot{\mathbf{x}}_B$ as

$$\begin{aligned}\begin{bmatrix} \boldsymbol{\tau} \\ \ddot{\mathbf{x}}_B \end{bmatrix} &= \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{M}}_{12} \\ \mathbf{0} & -\tilde{\mathbf{M}}_{22} \end{bmatrix}^{-1}\begin{bmatrix} \tilde{\mathbf{M}}_{11}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}_1 + \tilde{\mathbf{g}_1} - \tilde{\mathbf{J}}_{c1}^T\mathbf{F}_{ext} \\ \tilde{\mathbf{M}}_{21}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}_2 + \tilde{\mathbf{g}_2} - \tilde{\mathbf{J}}_{c2}^T\mathbf{F}_{ext} \end{bmatrix}\\&= \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{M}}_{12}\tilde{\mathbf{M}}_{22}^{-1} \\ \mathbf{0} & -\tilde{\mathbf{M}}_{22}^{-1} \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{M}}_{11}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}_1 + \tilde{\mathbf{g}_1} - \tilde{\mathbf{J}}_{c1}^T\mathbf{F}_{ext} \\ \tilde{\mathbf{M}}_{21}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}_2 + \tilde{\mathbf{g}_2} - \tilde{\mathbf{J}}_{c2}^T\mathbf{F}_{ext} \end{bmatrix} \quad (10)\end{aligned}$$

using the inversion formula of a block matrix. By extracting the joint torque vector, we have the inverse dynamics control law

$$\begin{aligned}\boldsymbol{\tau} &= (\tilde{\mathbf{M}}_{11} - \tilde{\mathbf{M}}_{12}\tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21})\ddot{\mathbf{q}}\\&\quad + (\tilde{\mathbf{C}}_1 + \tilde{\mathbf{g}}_1 - \tilde{\mathbf{J}}_{c1}^T\mathbf{F}_{ext}) - \tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21}(\tilde{\mathbf{C}}_2 + \tilde{\mathbf{g}}_2 - \tilde{\mathbf{J}}_{c2}^T\mathbf{F}_{ext})\\&= (\tilde{\mathbf{M}}_{11} - \tilde{\mathbf{M}}_{12}\tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21})\ddot{\mathbf{q}}\\&\qquad + \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21} \end{bmatrix}\left(\tilde{\mathbf{C}} + \tilde{\mathbf{g}} - \tilde{\mathbf{J}}_c^T\mathbf{F}_{ext}\right) \quad (11)\end{aligned}$$

Note that equation (11) can also be considered as the extracted joint space dynamics with external forces. If we compare (5) and (11), the following relationships can be extracted:

$$\left(\mathbf{S}\tilde{\mathbf{M}}^{-1}\mathbf{S}^T\right)^{-1} = \tilde{\mathbf{M}}_{11} - \tilde{\mathbf{M}}_{12}\tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21} \quad (12)$$

$$\bar{\mathbf{S}}^T = \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21} \end{bmatrix}. \quad (13)$$

These these equations correspond to another representation of the joint space inertia matrix, Coriolis and gravity vectors [19]:

$$\mathbf{M} = \left(\mathbf{S}\tilde{\mathbf{M}}^{-1}\mathbf{S}^T\right)^{-1} = \tilde{\mathbf{M}}_{11} - \tilde{\mathbf{M}}_{12}\tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21} \quad (14)$$

$$\mathbf{C} = \bar{\mathbf{S}}^T\tilde{\mathbf{C}} = \tilde{\mathbf{C}}_1 - \tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21}\tilde{\mathbf{C}}_2 \quad (15)$$

$$\mathbf{g} = \bar{\mathbf{S}}^T\tilde{\mathbf{g}} = \tilde{\mathbf{g}}_1 - \tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21}\tilde{\mathbf{g}}_2. \quad (16)$$

Thus, we can reformulate the inverse dynamics control law with computed constraint forces from using $\boldsymbol{\tau}_{t-1}$ in equation (7) in a computationally more efficient representation:

$$\begin{aligned}\boldsymbol{\tau} &= (\tilde{\mathbf{M}}_{11} - \tilde{\mathbf{M}}_{12}\tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21})\ddot{\mathbf{q}}\\&\quad + \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{M}}_{22}^{-1}\tilde{\mathbf{M}}_{21} \end{bmatrix}\left(\tilde{\mathbf{C}} + \tilde{\mathbf{g}} - \tilde{\mathbf{J}}_c^T\boldsymbol{\lambda}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}, \boldsymbol{\tau}_{t-1})\right) \quad (17)\end{aligned}$$

For an actual robot implementation, we need to address how to efficiently extract all necessary quantities for this control law, as normal recursive Newton-Euler formulations of rigid body dynamics do not provide direct access to the inertia, Coriolis/centripetal, and gravity terms. We approach this issue in the framework of Featherstone's spatial algebra approach [17].

A. Efficient Computation of Rigid Body Dynamics Components

Featherstone [17] suggested various methods for computing forward and inverse versions of rigid body dynamics using spatial algebra. The following paragraphs clarify and augment some of Featherstone's original work, such that it becomes suitable for our control framework—interested readers may refer to [17] for detailed formulations of the robot dynamics with spatial algebra.

In [17], Section 7.5, it is stated that in spatial notation, the floating base rigid body dynamics can be written as:

$$\begin{bmatrix} \mathbf{M} & \hat{\mathbf{K}}_B^S \\ \hat{\mathbf{K}}_B & \hat{\mathbf{I}}_{0,B}^C \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{q}} \\ \hat{\mathbf{a}}_{0,B} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} - \mathbf{C} \\ -\hat{\mathbf{f}}_{0,B} \end{bmatrix} \quad (18)$$

Here, variables with ^ superscripts denote spatial variables as defined in [17], and we also rearranged the matrix/vector components in a compatible notation to the previous sections of this paper. Moreover, we assume that there are no external forces acting on the base of the robot. In (18), $\mathbf{Q}$ is the joint torque vector, $\mathbf{C}$ includes the Coriolis and gravity components, $\hat{\mathbf{I}}_{0,B}^C$ is the composite-rigid-body inertia of the base link, $\hat{\mathbf{f}}_{0,B}$ is the bias force required to give the base link zero acceleration, and $\hat{\mathbf{K}}_B$ is the $6 \times n$ matrix whose i^{th} column is $\hat{\mathbf{I}}_{i,B}^C\ \hat{\mathbf{s}}_{i,B}$ where $\hat{\mathbf{s}}_{i,B}$ is the joint axis (see [17] for details).

All quantities in (18) can be computed from efficient algorithms in [17]. However, we need to extract the floating base inertia matrix in *non-spatial* notation from this equation. First, it should be noted that Equation (18) is derived in base coordinates, as indicated by the subscripts B on most variables, and not world coordinates, which we have been using in all our previous derivations. In spatial algebra, care needs to be taken in coordinate transformations as some spatial quantities assume very different values for different coordinate origins, and even numerical inaccuracies can be introduced by imprudent choices of coordinate systems [20]. A useful choice of a world coordinate system is at the center of the base link of the robot, denoted by W_B—this origin is identical with that of the base coordinate system, but the coordinate system's orientation is now aligned with the world coordinate system. Thus, the spatial coordinate transformation from base to world coordinates becomes

$$ {}_{W_B}\hat{\mathbf{X}}_B = \begin{bmatrix} {}_W\mathbf{R}_B^{-1} & \mathbf{0} \\ \mathbf{0} & {}_W\mathbf{R}_B^{-1} \end{bmatrix} \tag{19} $$

where ${}_W\mathbf{R}_B$ is the rotation matrix from base to world coordinates. Equation (18) can then be expressed in world coordinates as:

$$ \begin{bmatrix} \mathbf{M} & \hat{\mathbf{K}}^S_{W_B} \\ \hat{\mathbf{K}}_{W_B} & \hat{\mathbf{I}}^C_{0,W_B} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \hat{\mathbf{a}}_{0,W_B} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}-\mathbf{C} \\ -\hat{\mathbf{f}}_{0,W_B} \end{bmatrix} \tag{20} $$

where

$$ \hat{\mathbf{K}}^S_{W_B} = \hat{\mathbf{K}}^S_B \; {}_B\hat{\mathbf{X}}_{W_B} \tag{21} $$

$$ \hat{\mathbf{K}}_{W_B} = {}_{W_B}\hat{\mathbf{X}}_B \; \hat{\mathbf{K}}_B \tag{22} $$

$$ \hat{\mathbf{I}}^C_{0,W_B} = {}_{W_B}\hat{\mathbf{X}}_B \; \hat{\mathbf{I}}^C_{0,B} \; {}_B\hat{\mathbf{X}}_{W_B} \tag{23} $$

$$ \hat{\mathbf{a}}_{0,W_B} = {}_{W_B}\hat{\mathbf{X}}_B \; \hat{\mathbf{a}}_{0,B} \tag{24} $$

$$ \hat{\mathbf{f}}_{0,W_B} = {}_{W_B}\hat{\mathbf{X}}_B \; \hat{\mathbf{f}}_{0,B} \tag{25} $$

By inserting the relationship between spatial accelerations and conventional accelerations

$$ \hat{\mathbf{a}}_{0,W_B} = \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \ddot{\mathbf{r}}_{W_B} + \dot{\mathbf{r}}_{W_B} \times \boldsymbol{\omega} \end{bmatrix} \tag{26} $$

into (20) and rearranging it, we obtain

$$ \begin{bmatrix} \mathbf{M} & \hat{\mathbf{K}}^S_{W_B} \\ \hat{\mathbf{K}}_{W_B} & \hat{\mathbf{I}}^C_{0,W_B} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \dot{\boldsymbol{\omega}} \\ \ddot{\mathbf{r}}_{W_B} \end{bmatrix} \tag{27} $$

$$ = \begin{bmatrix} \mathbf{Q} \\ \mathbf{0} \end{bmatrix} - \left(\begin{bmatrix} \mathbf{C} \\ \hat{\mathbf{f}}_{0,B_W} \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{K}}^S_{W_B} \\ \hat{\mathbf{I}}^C_{0,W_B} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{r}}_{B_W} \times \boldsymbol{\omega} \end{bmatrix} \right). \tag{28} $$

Thus, for conventional notation in world coordinates, we can extract the inertia matrix $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{C}} + \tilde{\mathbf{g}}$:

$$ \tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M} & \hat{\mathbf{K}}^S_{W_B} \\ \hat{\mathbf{K}}_{W_B} & \hat{\mathbf{I}}^C_{0,W_B} \end{bmatrix} \tag{29} $$

$$ \tilde{\mathbf{C}} + \tilde{\mathbf{g}} = \left(\begin{bmatrix} \mathbf{C} \\ \hat{\mathbf{f}}_{0,B_W} \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{K}}^S_{W_B} \\ \hat{\mathbf{I}}^C_{0,W_B} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{r}}_{B_W} \times \boldsymbol{\omega} \end{bmatrix} \right) \tag{30} $$

where it needs to be noted that in [17], the gravity term is automatically included in the right hand side of (18).

Note that if there is no need to extract $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{C}} + \tilde{\mathbf{g}}$ terms and compute the constraint forces $\boldsymbol{\lambda}$ using them, there are computationally more efficient recursive formulations for floating base inverse dynamics based on the articulated body hybrid method and the composite rigid body hybrid method [17][1]. However, the control law (17) requires an extraction of $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{C}} + \tilde{\mathbf{g}}$ to compute the constraint forces $\boldsymbol{\lambda}$. Thus, once these quantities are available, the algebraic form (17) seems to be simpler than implementing the recursive floating inverse dynamics formulae with external forces.

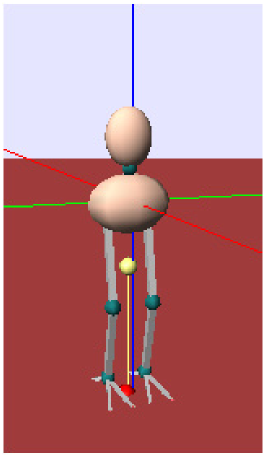


Fig. 2. Testbed: a 16 DOF simulated biped robot model.

IV. Evaluations

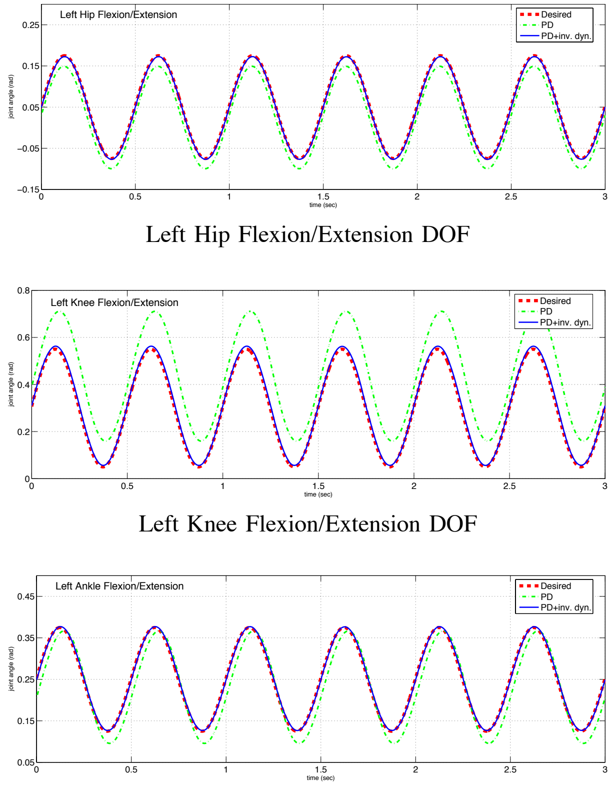
In this section, we evaluate the suggested approach on a simulated biped robot model as an example of a floating base system with constraints as shown in Figure 2. This simulated biped robot has 14 DOFs, where each leg has 7 DOF and the torso has 2 DOFs. We consider a robot model with the size and weight similar to a human lower body (the height is about 1m and the weight is approximately 40kg).

First, we consider joint space tracking control, and then we use the suggested approach as a part of an acceleration based task space control which requires a proper inverse dynamics control law.

1) Joint Space Tracking Control: As a test movement of joint space tracking control, we consider a squatting movement generated by simple sinusoidal joint trajectories, and compare the tracking performance of a low gain PD controller and a PD controller+inverse dynamics with constraint forces. These constraint forces are computed with an assumption that both feet are fixed to the ground, and used in the inverse dynamics control law.

To generate squatting movements, we used sinusoid desired joint trajectories for the hip, knee and ankle joints of both legs (slow 1Hz, fast 2Hz), and the desired position for the rest of the DOFs were kept constant. Figure 3 shows a representative comparison of the tracking results using only a low gain PD controller and the suggested inverse dynamics control with constraint forces. Plots of the major degrees of freedom (hip, knee and ankle flexion/extension DOFs) for the fast squatting movement at 2Hz are shown in Figure 3. The result indicates that with the low gain PD controller, the robot could not track the desired trajectory very well. However, in contrast, the inverse dynamics control law with constraint forces achieved much improved tracking performance by compensating the nonlinearity of the robot dynamics and the contact forces. Tracking errors (measured by the L_2 norm defined as $L_2[\mathbf{e}(t)] = \sqrt{(1/T)\int_{t_0}^{t_f} ||\mathbf{e}(t)||^2 dt}$ where $\mathbf{e}(t) = \mathbf{q}_d - \mathbf{q}$ and $T = t_f - t_0$) using the PD controller were 1.82×10^{-1} (slow movement) and 2.04×10^{-1} (fast

[1] The original formulations in [17] do not include external forces, but it can be modified to incorporate external forces $\mathbf{F}_{ext}$ by inserting them in the appropriate places.



Left Hip Flexion/Extension DOF

Left Knee Flexion/Extension DOF

Left Ankle Flexion/Extension DOF

Fig. 3. Comparison of the tracking results of the three major degrees of freedom used for the fast squatting movement (2Hz) with the simulated biped. With a low gain PD controller, the robot could not track the desired trajectory very well. However, the inverse dynamics control law with constraint forces achieves much improved tracking.

movement), and using the inverse dynamics control law, the L_2 norm of the tracking errors were 1.40×10^{-2} (slow movement) and 1.63×10^{-2} (fast movement), respectively. In particular, a significant improvement can be seen in the tracking error of the knee joint whose movement is much influenced by the effect of the gravity as illustrated in the middle of Figure 3. The results suggest the effectiveness of the inverse dynamics control law by compensating the nonlinearities of the robot dynamics and contact forces.

2) Application to Task Space Control: In our previous work [2], we explored task space control with a fixed base redundant robot arm. Our empirical evaluations demonstrated that the simplified acceleration based control was the most promising approach among several different methods. In order to apply such a task space control approach to a biped robot, we need a proper inverse dynamics control law including contact constraints.

In this section, we consider a task of controlling the base link position $\mathbf{x}_B$ (located at the center of the pelvis, see Figure 1) of the simulated biped robot model with an extension of the simplified acceleration based task space control method which we explored in [2]. In this paper, we consider a slight variant formulation of the simplified acceleration based task space control. First, for a given task space acceleration command $\ddot{\mathbf{x}}_d$, the desired joint accelerations are computed

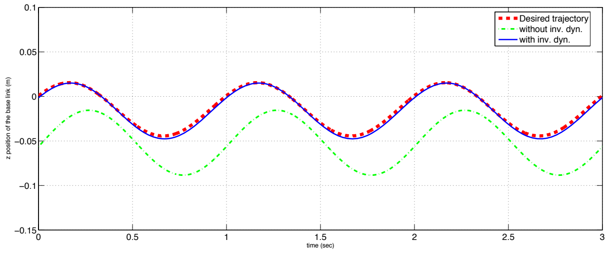


Fig. 4. Comparison of the tracking results of the task space control with inverse dynamics (32) and without inverse dynamics (33) using the same gain settings. The plot shows the tracking result of the vertical motion of the base link (pelvis) of the robot for a given sinusoidal desired trajectory to achieve squatting movement. Without inverse dynamics, the robot could not achieve good tracking performance largely due to undercompensation of the gravity terms. In contrast, with the inverse dynamics, tracking performance was much improved.

using the Jacobian pseudoinverse as

$$\ddot{\mathbf{q}}_d = \mathbf{J}^{\dagger}(\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}\dot{\mathbf{q}}) \tag{31}$$

where $\mathbf{J}$ denotes the task space Jacobian matrix taking the contact constraints into account. Then, an inverse dynamics control law with the addition of task space PD error terms and Liégeois-like null space projection term with damping is used to obtain the joint torque command

$$\begin{aligned}\boldsymbol{\tau} = &\, Invdyn(\ddot{\mathbf{q}}_d, \dot{\tilde{\mathbf{q}}}, \tilde{\mathbf{q}}, \boldsymbol{\lambda}(\dot{\tilde{\mathbf{q}}}, \tilde{\mathbf{q}}, \boldsymbol{\tau}_{t-1})) + \mathbf{J}^{\dagger}(\mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) \\ &+ (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})(-\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\mathbf{K}_w(\mathbf{q}_{rest} - \mathbf{q}))\end{aligned} \tag{32}$$

where $\dot{\mathbf{e}}$ and $\mathbf{e}$ are the velocity and position errors in the task space, respectively, $\mathbf{K}_d$ and $\mathbf{K}_p$ are positive definite gain matrices, $\mathbf{K}_{q,d}$ is the damping gain matrix, α is a positive scaling factor, $\mathbf{K}_w$ is the weighting matrix, and $\mathbf{q}_{rest}$ is some rest posture. As a benchmark movement, we consider a task of tracking squatting movement specified by a sinusoidal desired trajectory at 1Hz in the vertical direction for the pelvis position. We demonstrate the effectiveness of the suggested inverse dynamics control by comparing to the control law without *invdyn*() component in (32) with the same task space gain settings as

$$\begin{aligned}\boldsymbol{\tau} = &\, \mathbf{J}^{\dagger}(\mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) \\ &+ (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})(-\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\mathbf{K}_w(\mathbf{q}_{rest} - \mathbf{q})).\end{aligned} \tag{33}$$

Note that this controller does not compensate the nonlinearities of the robot dynamics. Figure 4 shows the comparison of the tracking results of the movement of the pelvis as task space coordinates in the vertical direction. Without inverse dynamics, the robot could not achieve good tracking to the desired trajectory largely due to undercompensation of the gravity terms. However, in contrast, the inverse dynamics control law with constraint forces achieved improved tracking performance by compensating the nonlinearity of the dynamics. The L_2 norm of the tracking error of the task space coordinates is 3.81×10^{-3} with inverse dynamics and 8.60×10^{-2} without the inverse dynamics, respectively. The results demonstrate the effectiveness of the suggested inverse dynamics approach, however, still further investigation will be necessary in order to achieve more complex movements including balancing, transition of contact constraints and combination with joint space pattern generator in a hierarchical manner.

V. Conclusion

In this paper, we presented an approximated way of computing inverse dynamics of a free floating base system with contact constraint. We formulated the constrained inverse dynamics based on the idea of Featherstone's hybrid dynamics algorithm incorporating external forces. We also presented how the necessary components to implement the control law can be computed from Featherstone's efficient rigid body dynamics formulation based on spatial algebra.

We empirically evaluated our approach on a simulated biped robot model. We achieved improvement of tracking performance with our floating inverse dynamics approach compared to a PD controller suggesting its effectiveness. We also evaluated the effectiveness of the suggested approach on the acceleration based task space control framework which require floating base inverse dynamics with contact constraints.

Future work will address implementation of the suggested approach on an actual humanoid robot, comparison with other constraint control approaches, transition of contact constraints, learning and prediction of contact forces, and investigation of the principle of internal model based biological motor control.

Acknowledgments

We would like to thank Prof. Roy Featherstone for clarifications of spatial algebra. This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, ECS-0326095, ANI-0224419, the DARPA program on Learning Locomotion, a NASA grant AC#98-516, an AFOSR grant on Intelligent Control, the ERATO Kawato Dynamic Brain Project funded by the Japan Science and Technology Agency, and the ATR Computational Neuroscience Laboratories.

References

[1] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[2] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Comparative experiments on task space control with redundancy resolution," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3522–3529.

[3] ——, "Operational space control: A theoretical and empirical comparison," *International Journal of Robotics Research*, (submitted).

[4] O. Khatib, O. Brock, K. Chang, D. Ruspini, L. Sentis, and S. Viji, "Human-centered robotics and interactive haptic simulation," *International Journal of Robotics Research*, vol. 23, no. 2, pp. 167–478, 2004.

[5] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 4, pp. 505–518, 2005.

[6] J. Peters, M. Mistry, F. Udwadia, R. Cory, J. Nakanishi, and S. Schaal, "A unifying methodology for the control of robotic systems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3522–3529.

[7] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 79–91, 2004.

[8] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[9] S. Arimoto, *Control Theory of Non-linear Mechanical Systems: A Passivity-based and Circuit-theoretic Approach*. Oxford University Press, 1996.

[10] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.

[11] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. MIT Press, 1990.

[12] V. De Sapio and O. Khatib, "Operational space control of multibody systems with explicit holonomic control," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2005, pp. 2961–2967.

[13] J. Park and O. Khatib, "Multi-link multi-contact force control for manipulators," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2005, pp. 3624–3629.

[14] A. Greenfield, U. Saranli, and A. A. Rizzi, "Solving models of controlled dynamic planar rigid-body systems with frictional contact," *International Journal of Robotics Research*, vol. 24, no. 11, pp. 911–931, 2005.

[15] J. Park and O. Khatib, "Contact consistent control framework for humanoid robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2006, pp. 1963–1969.

[16] S.-H. Hyon and G. Cheng, "Gravity compensation and whole-body force interaction for humanoid robots," *IEEE-RAS International Conference on Humanoid Robots*, 2006.

[17] R. Featherstone, *Robot Dynamics Algorithms*. Kluwer Academic Publishers, 1987.

[18] F. E. Udwadia and R. E. Kalaba, *Analytical Dynamics: A New Approach*. Cambridge University Press, 1996.

[19] L. Sentis and O. Khatib, "Control of free-floating humanoid robots through task prioritization," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2005, pp. 1730–1735.

[20] R. Featherstone, (personal communication).

Appendix

A. Spatial Algebra Notation

In this section, we briefly describe spatial algebra notation used in Featherstone's rigid body dynamics algorithm [17].

Spatial velocity definition:

$$\hat{\mathbf{v}} = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v} + \mathbf{r} \times \boldsymbol{\omega} \end{bmatrix} \tag{34}$$

where $\mathbf{v}$ is the translatory velocity of a point P on the body, $\mathbf{r}$ is the offset vector OP from the origin O of the coordinate system to P, and $\boldsymbol{\omega}$ is the regular angular velocity.

Spatial acceleration (component-wise time derivative of spatial velocity above):

$$\hat{\mathbf{a}} = \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \ddot{\mathbf{r}} + \dot{\mathbf{r}} \times \boldsymbol{\omega} + \mathbf{r} \times \dot{\boldsymbol{\omega}} \end{bmatrix} \tag{35}$$

Spatial coordinate transformation:

$${}_P\hat{\mathbf{X}}_O = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{R}\,\mathbf{r}\times^T & \mathbf{R} \end{bmatrix} \tag{36}$$

$${}_O\hat{\mathbf{X}}_P = \begin{bmatrix} \mathbf{R}^{-1} & \mathbf{0} \\ \mathbf{r}\times\ \mathbf{R}^{-1} & \mathbf{R}^{-1} \end{bmatrix} \tag{37}$$

where $\mathbf{R}$ is the regular rotation matrix, and $\mathbf{r}$ is the offset vector between O and P, and

$$\mathbf{r}\times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \tag{38}$$

Spatial transpose is denoted by ^{S}, and generalized joint forces are denoted by $\mathbf{Q}$.